11. Shekhar

Program Name: Shekhar.java **Input File: shekhar.dat**

Shekhar just got the new and improved Computron-2021 graphing calculator, made by Lone Star State Devices. The Computron-2021 can do all mathematical calculations from solving algebraic equations to taking derivatives of multivariable functions. Being the top of the line calculator it is, Shekhar is bound and determined to try and find a bug with the calculator so he can report back to Lone Star State Devices in hopes of earning an academic scholarship.

Shekar believes the Computron-2021 doesn't ensure that the delimiters of a mathematical expression are nested correctly. The Computron-2021 uses parentheses ( ), brackets [ ], and braces { } as delimiters. For a mathematical expression to have correctly nested delimiters, the following two statements must be true:

1. There are an equal number of right and left delimiters
2. Every right delimiter is preceded by a matching left delimiter.

For example, the expression $7 - ((x * ((x + y) / (j - 3)) + y) / (4 - 2.5))$ is nested correctly because the two above conditions are satisfied. The expression $((a + b)$ is not nested correctly because there are two left parentheses but only one right parenthesis, thus causing statement 1) above to be not met. The expression $\{a * (b + c\})$ is not nested correctly either. Although there are equal numbers of right and left delimiters, the right brace } is preceded by a left parenthesis (, thus causing statement 2) not to be met.

Shekar needs your help writing a program that can be used to double check all the challenging, weird, and wonky expressions he plans on using to break the Computron-2021.

Input: The input begins with an integer I, the number of test cases. I is guaranteed to be in range [1,30]. The following I lines will contain a single mathematical expression. The expression may constitute integer literals, variables, addition, subtraction, multiplication, and division, but the only delimiters allowed are parentheses, brackets, and braces.

Output: For each mathematical expression, you are to output the expression followed by either "`is nested correctly`" or "`is nested incorrectly`"

Sample input:

```
12
7 - ((x * ((x + y) / (j - 3)) + y) / (4 - 2.5))
((a + b)
{a*(b+c})
7 - ({x * [(x + y) / (j - 3)] + y} / (4 - 2.5))
a + b(
)a + b(
{[(a + b) * 10 + c] * 10 + d}
((((((((((()))))))))))
((((((((((()))))))))))(
)((((((((((()))))))))))
```

Sample output:

```
7 - ((x * ((x + y) / (j - 3)) + y) / (4 - 2.5)) is nested correctly
((a + b) is nested incorrectly
{a*(b+c}) is nested incorrectly
7 - ({x * [(x + y) / (j - 3)] + y} / (4 - 2.5)) is nested correctly
a + b( is nested incorrectly
)a + b( is nested incorrectly
{[(a + b) * 10 + c] * 10 + d} is nested correctly
((((((((((()))))))))))) is nested correctly
((((((((((()))))))))))) is nested incorrectly
((((((((((()))))))))))) is nested incorrectly
((((((((((()))))))))))(( is nested incorrectly
)((((((((((()))))))))))) is nested incorrectly
```